NCED Summer Institute 2017

Introduction to programming bootcamp

August 12, 2017

Mariela Perignon, CSDMS Integration Facility, CU Boulder
perignon@colorado.edu

Requirements: No programming experience is expected. Please collaborate and help your neighbors during this workshop.

Setup on Cloud9

To avoid the hassles of installing software in individual computers, we are going to use the Cloud9 IDE to explore Python.

Cloud9 is an online integrated development environment that supports a wide variety of programming languages. It provides developers with pre-configured workspaces in Ubuntu containers via Docker. Cloud9 is owned by Amazon and hosted on Google Compute Engine.

Joining our Cloud9 team:

You should have received an email this morning inviting you to join our Cloud9 team. **Ask for help now if you did not receive this email.**

Already have a Cloud9 account and didn't get the email?
If the email address you used for NCED is also the email for your Cloud9 account, you *might* have been automatically added to the team. Log into Cloud9 at https://c9.io. The team "NCED summer institute 2017" should be listed in your Dashboard under Your Team Subscriptions. If it's not there, ask for a new invitation.

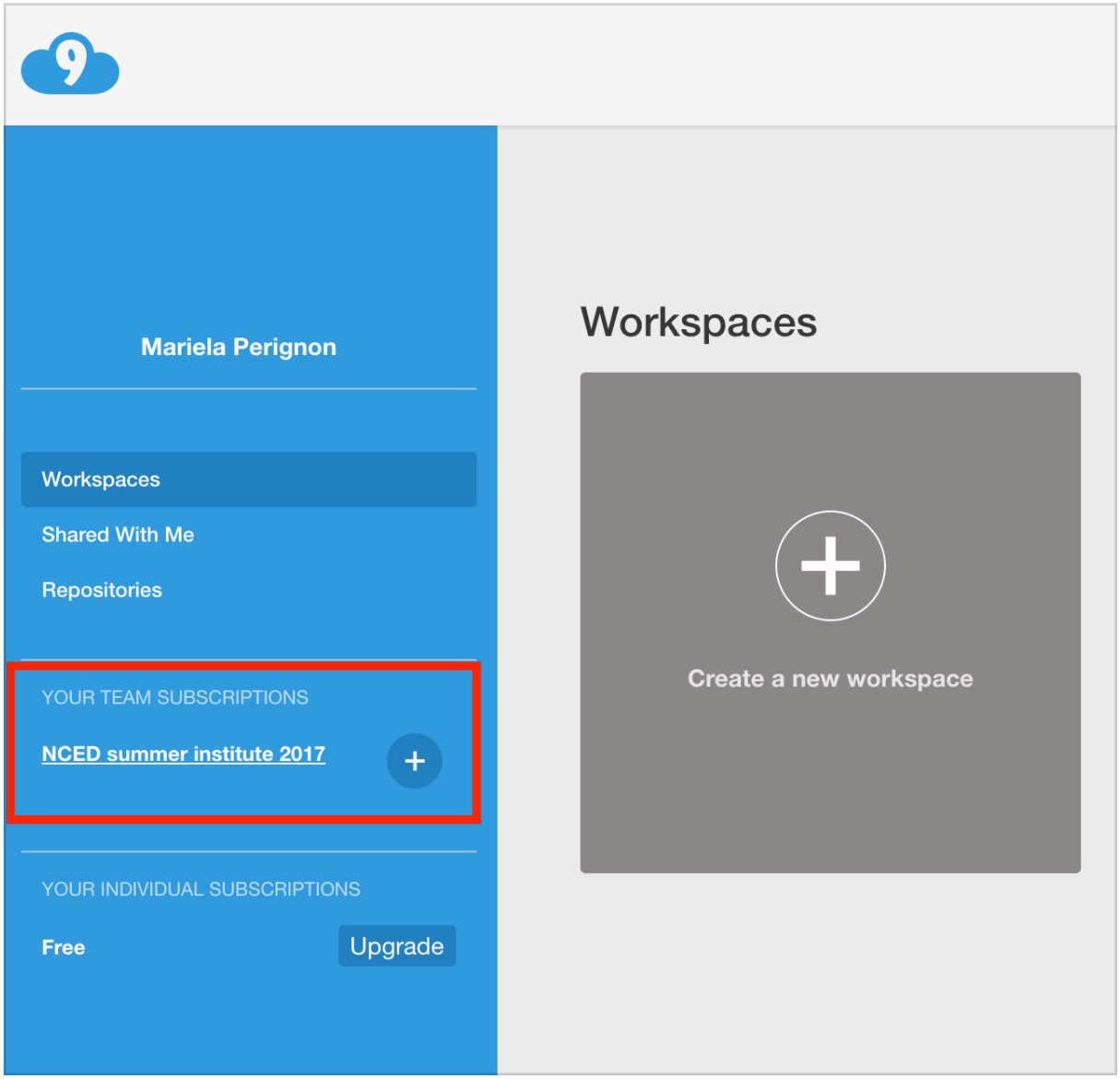
The email should look like this:

- Email sender: support@c9.io
- Subject: Team nced2017 has invited you to join their team on Cloud9

- Follow the link in the email to log into Cloud9 and join our team:
- If you do not have an account on Cloud9, click on "Create new account" and follow the instructions.

- If you already have an account, click on “Sign in”.

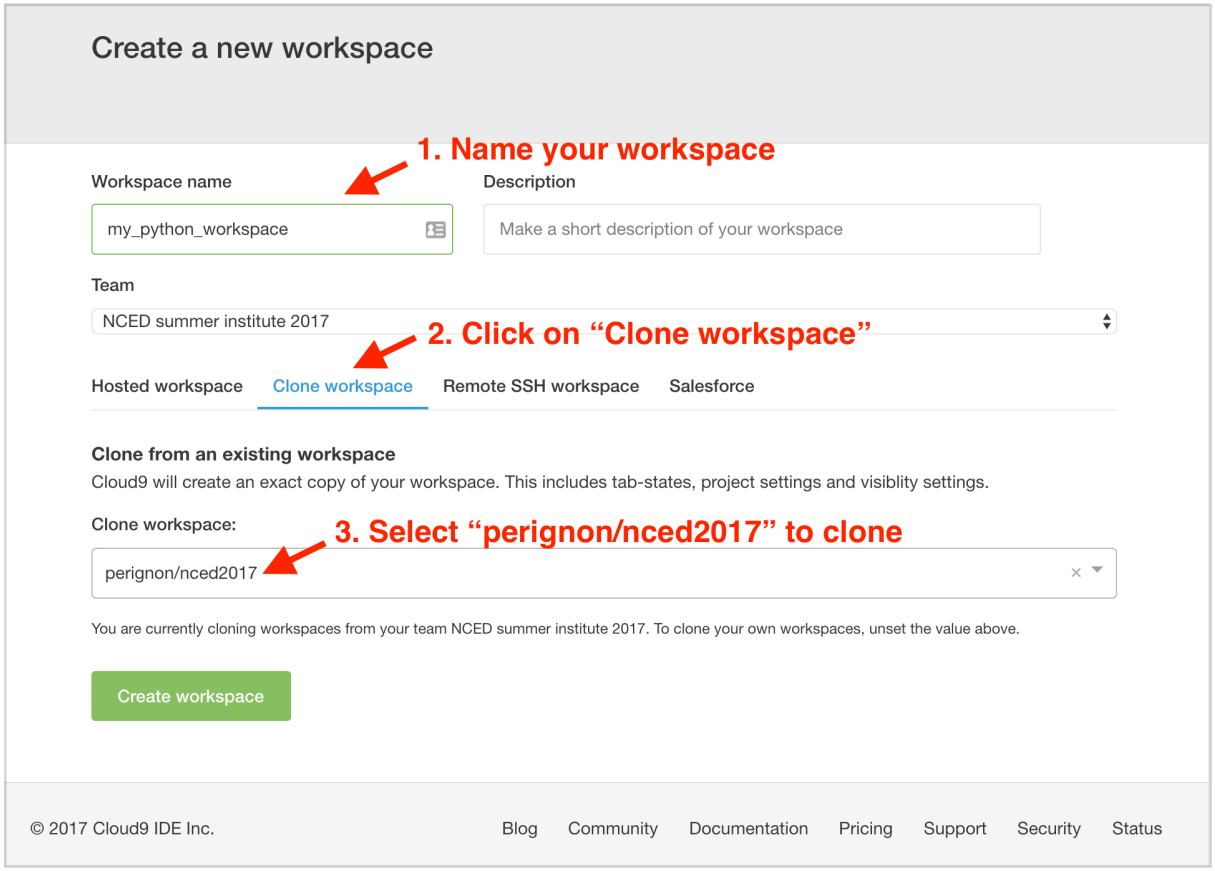
  This is your Cloud9 Dashboard. The team “NCED summer institute 2017” is listed under Your Team Subscriptions on the left side:



- Go back to your email client and look for a welcome email from support@c9.io. Follow the link to set a password for your account. This will let you jump back in if you accidentally close the browser!

Creating a workspace:

- In your Dashboard, click on “Create a new workspace” and follow the instructions below:



This process creates a new Ubuntu workspace for your user that is identical to an existing workspace where all the necessary software and files is pre-installed.

- **Dismiss the "Disk is full" warnings.** We added a script to the login process that cleans up files left over from software installation. The 2GB of disk space allocated to your new workspace should be only half full.

Using Python in the command line:

- In the Cloud9 menu, go to Window ➜ New Terminal to open a new Bash terminal window.

What is Bash?

Early computers were programmed by connecting wires in particular configurations. Modern computers, on the other hand, receive commands through a **graphical user interface** (*GUI*) and software. Between the 1950s to the 1980s, however, most people interacted with computers through line printers. These devices only allowed input and output of text, so programming languages and software interfaces had to be designed around that constraint. This kind of interface is called a **command-line interface** (*CLI*)

We use the CLI through a program called the **command shell** . The user types into the shell, which then figures out what commands to run and orders the computer to execute them. The shell is called "the shell" because it encloses the operating system to hide some

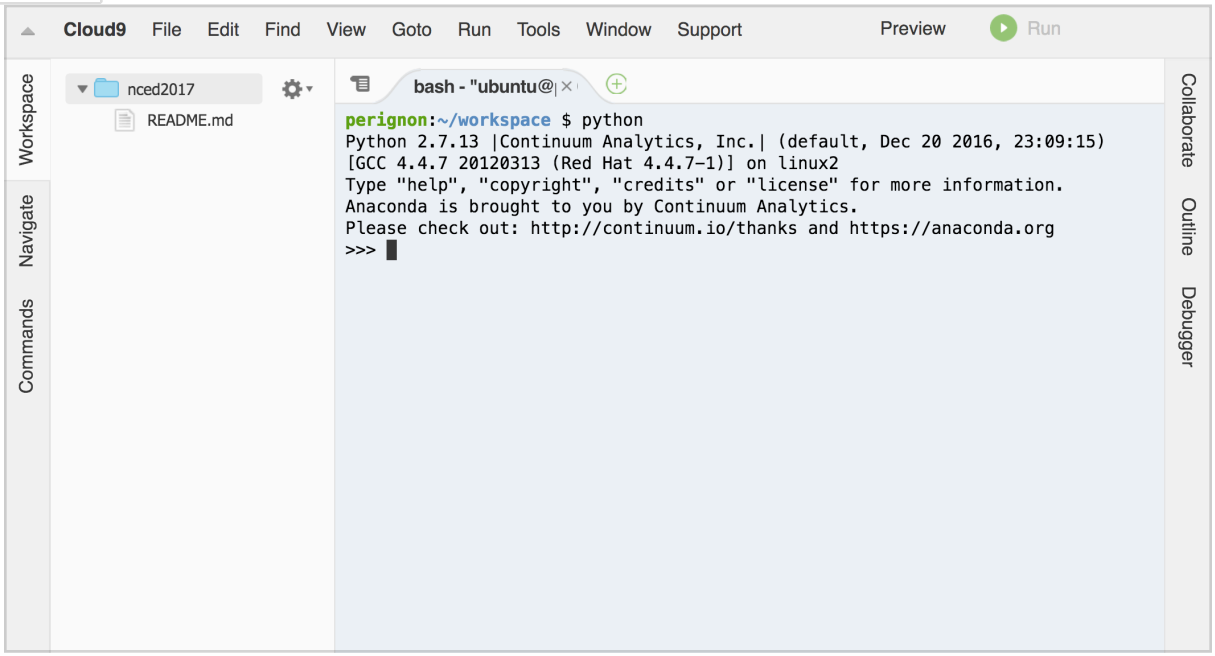
of its complexity and protects it from the user.

The most popular Unix shell is **Bash**. Bash is the default shell on most modern implementations of Unix and in most packages that provide Unix-like tools for Windows.

You can learn more about using Bash by following these lessons.

- To start Python, type:

  `python`



  Notice that the **prompt** inside Python changed from `$` to `>>>`.

- Try typing some simple commands and look at the output:

  `2 + 2`

  `print “Hello, world!”`

- To exit the Python environment, type:

  `quit()`

  You can also exit Python by typing `control-D`, an End-of-Transmission (EOF) character.

What is Anaconda?

The power of Python comes from the tens of thousands of libraries that can be used to extend its capabilities. Installing Python and its libraries by hand, however, can be very complicated. Anaconda is a Python distribution that comes pre-packages with the libraries that are most useful for scientific computing. It also includes *conda*, a package manager

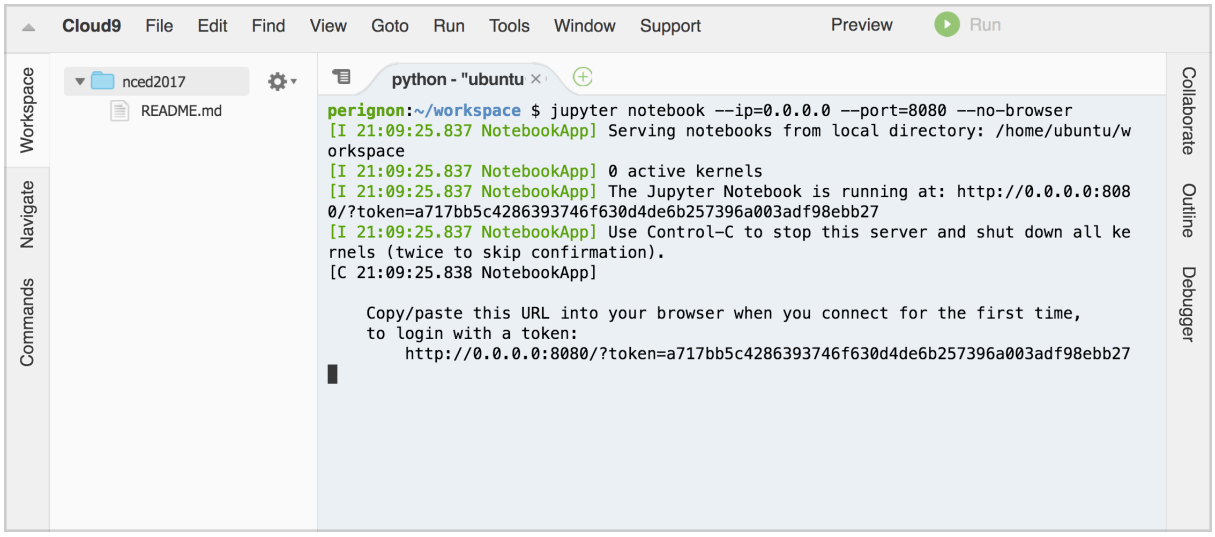
that can be used to find and install other Python libraries.

Starting a Jupyter Notebook:

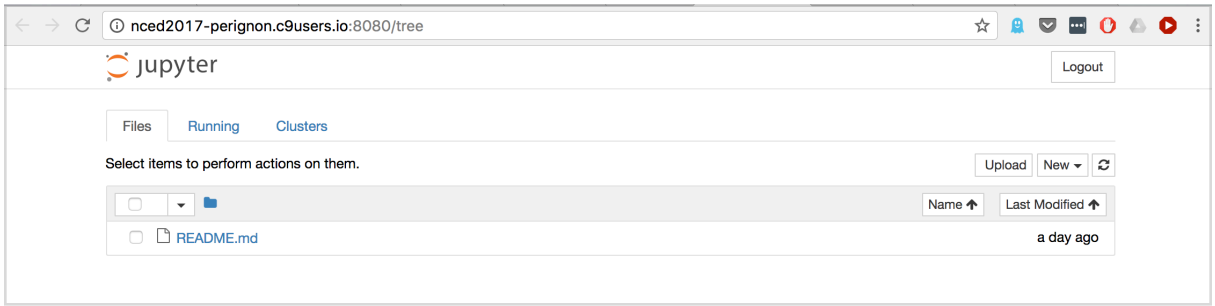
It can be difficult to keep track of inputs and output when using Python through the command-line. Instead of typing individual commands into the shell, developers most often write their code in a text file (saved with a `.py` extension) and run the file from the command line.

A more effective way to learn to program in Python is to use an interactive environment like **Jupyter Notebook**. Jupyter Notebook is an open-source web application for creating and sharing documents that contain live code, equations, visualizations and explanatory text.

- To start the Jupyter Notebook server, type in the terminal window:
  `jupyter notebook --ip=0.0.0.0 --port=8080 --no-browser`
  (We only need to specify these options because we are using Cloud9!)



- Click on the URL at the end of the text and select *Open*. This will open a new browser tab showing the Jupyter dashboard:



It didn't work!!!

For added security, the Jupyter Notebook server requires an authentication token to run for the first time. The spotty WIFI connection of the Guest network can cause issues with authentication and prevent the browser from loading the page. If that happens, try to load it by hand:

- Edit the URL to keep only the root domain:
`https://<name-of-workspace>-<your-username>.c9users.io`

- If it shows a page asking for an authentication token, go back to your workspace and copy the string of characters after (and including) `:8080`. Each Jupyter notebook server has its own individual authentication token, so you must **copy the token from your own terminal window**:

```
Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://0.0.0.0:8080/?
token=772d717017233ec7b50a38a31a03f55f77b1e5eafd7dc2cb
```

Add it to the URL so it looks like this:
`https://<name-of-workspace>-<your-username>.c9users.io:8080/?token=<your-token>`

The browser should log into the notebook server and open the Jupyter dashboard.

Access the Python lessons here: Software Carpentry: Programming with Python.